An Interface Transformation Strategy for AF-IPPS

Albuquerque, Ronald J.
Hockert, Christopher W.
San Antonio, Ralph C.

Sponsor: US Air Force
Department No.: E141
Contract No.:
Project No.: 03124AH0-D1

Document No: WN120134
Bedford, Massachusetts

MITRE

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.		
1. REPORT DATE **DEC 2012**	2. REPORT TYPE	3. DATES COVERED **00-00-2012 to 00-00-2012**
4. TITLE AND SUBTITLE **An Interface Transformation Strategy for AF-IPPS**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The MITRE Corp,202 Burlington Road,Bedford,MA,01730-1420**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT **The Air Force Integrated Personnel and Pay System (AF-IPPS) Program is intended to replace the systems currently used for AF Active Duty, Reserve and Air National Guard components; AF-IPPS will replace the Military Personnel Data System (MilPDS) and the Defense Joint Military (Pay) System (DJMS) which currently perform these functions. The core of AF-IPPS will be implemented using an Enterprise Resource Planning (ERP) platform such as PeopleSoft, Oracle, or SAP. Like other AF and Department of Defense (DoD) ERP programs, AF-IPPS is expected to have 100+ interfaces with 90+ external trading partners, using approximately 640 uniquely defined data transactions. Experience with other DOD ERP programs indicates that a high number of interfaces pose technical, cost and schedule risk to the program. The purpose of the interface strategy prototype effort was to mitigate this risk by demonstrating the applicability of modern messaging and communication approaches to insulate the ERP from interface changes by the legacy applications, allowing them to gradually evolve to the new communications and data standards with minimal impact to AF-IPPS. The prototype development effort leveraged information content from the predecessor Defense Integrated Military Human Resource System (DIMHRS) program along with actual AF-IPPS planned interface content. Using this information, the prototype effort began with the eXtensible Markup Language (XML) definition of a person object. Subsequent effort included both application development and the integration of open-source and Commercial-Off-the-Shelf (COTS) applications. The result was a prototype AF-IPPS translation layer that successfully implements a publish/subscribe interface model. While the translation layer was not tested for performance it is built with software components that are widely used in the software industry and could thus be enhanced to meet specific performance requirements. A next step for prototype is integration with the Enterprise Level Security (ELS) service provided by the Air Force. This paper describes the interface prototype design and the details of both the person object definition and the translation layer implementation.**		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Table of Contents

List of Figures

List of Tables

Abstract

The Air Force Integrated Personnel and Pay System (AF-IPPS) Program is intended to replace the systems currently used for AF Active Duty, Reserve and Air National Guard components; AF-IPPS will replace the Military Personnel Data System (MilPDS) and the Defense Joint Military (Pay) System (DJMS) which currently perform these functions. The core of AF-IPPS will be implemented using an Enterprise Resource Planning (ERP) platform such as PeopleSoft, Oracle, or SAP.

Like other AF and Department of Defense (DoD) ERP programs, AF-IPPS is expected to have 100+ interfaces with 90+ external trading partners, using approximately 640 uniquely defined data transactions. Experience with other DOD ERP programs indicates that a high number of interfaces pose technical, cost and schedule risk to the program.

The purpose of the interface strategy prototype effort was to mitigate this risk by demonstrating the applicability of modern messaging and communication approaches to insulate the ERP from interface changes by the legacy applications, allowing them to gradually evolve to the new communications and data standards with minimal impact to AF-IPPS.

The prototype development effort leveraged information content from the predecessor Defense Integrated Military Human Resource System (DIMHRS) program along with actual AF-IPPS planned interface content. Using this information, the prototype effort began with the eXtensible Markup Language (XML) definition of a person object. Subsequent effort included both application development and the integration of open-source and Commercial-Off-the-Shelf (COTS) applications. The result was a prototype AF-IPPS translation layer that successfully implements a publish/subscribe interface model. While the translation layer was not tested for performance it is built with software components that are widely used in the software industry and could thus be enhanced to meet specific performance requirements.

A next step for prototype is integration with the Enterprise Level Security (ELS) service provided by the Air Force. This paper describes the interface prototype design and the details of both the person object definition and the translation layer implementation.

Program Overview

The AF-IPPS program will deliver an integrated personnel and pay capability to support active duty, AF Reserve and Air National Guard units. AF-IPPS represents the AF commitment to modernizing business practices and to providing enhanced Total Force support to combatant commanders, war-fighters, and their families. “Integrated” means that changes in a person’s status, once entered into the system, will result in the proper pay calculation and disbursement without additional data entry. In addition, AF-IPPS will provide active duty, Guard and Reserve airmen with a single record of their service for their entire career.

AF-IPPS will also implement a self-service capability that will allow airmen to update personal information and access their pay records on a 24 X 7 basis. The system will have many interfaces to external systems.

Lessons Learned: Interfaces as a Major Risk

There are many lessons-learned presentations and Government reports that document the particular problems that DOD ERP implementations have experienced with large numbers of interfaces that add both complexity and risk to the programs. As a result, DOD guidance to

implementation programs has been to strive to reduce the number of interfaces. Unfortunately, short of just unplugging systems, this is very difficult guidance to implement. It requires extensive business process reengineering; considerable change management; and strong, committed leadership.

Background

AF-IPPS has prepared and coordinated Interface Control Agreements with all of its interface partners. These agreements document the responsibility and procedures for mutual notification of changes that are necessary and planned to occur to the information exchanges.

AF-IPPS trading partners currently execute these interfaces with legacy AF and Defense Finance & Accounting Services (DFAS) systems. They use different formats and protocols such as secure file transfer protocol (SFTP), web services, and Database-Link. Neither AF-IPPS nor many of its trading partners are funded to change the manner in which they interface with other systems. Therefore, AF-IPPS must be able to communicate with them using their existing technologies & protocols when the system goes live.

Additionally, the partners exchange or require different information although much is common to all. Consequently, many data elements are found in more than one exchange. Hence, it would be beneficial to publish AF-IPPS information once in a common format for all to obtain those elements that they need, resulting in fewer queries against the transactional system. This "publish/subscribe" approach requires a translation of the data into existing partner formats. The translation layer depicted in this prototype architecture is a way to satisfy this requirement.

Air Force Use of Service-Oriented Architecture (SOA)

The DOD has made "services" part of the Net Centric Data Strategy. The tenets of this strategy include making data visible, accessible, understandable, trusted, interoperable, and responsive to user needs. Furthermore, the Air Force, according to ESCI 63-1200, directs that acquisition programs contribute data and participate in enterprise-related engineering analyses, evaluations, and/or assessments. This instruction also directs programs to provide visibility of web-services based software systems and applications developed by the program. In compliance with this instruction the AF-IPPS System Requirements Document (SRD) states the following: "*Use of Web Services should be the standard for data exchanges within the AF Manpower and Personnel domain, and the preferred solution for all other system partners.*"

The interface prototype implements SOA concepts using related technologies such as Simple Object Access Protocol (SOAP), Representational State Transfer (REST) and Java Enterprise Edition (Java EE) to implement a reusable "translation service." For SOAP and REST protocols, XML and JavaScript Object Notation (JSON) were used. In addition, a publish/subscribe architecture was incorporated to allow data to be published only once.

Prototype Design

The design phase commenced with the development of a canonical Person business object XML Schema Definition (XSD). This was achieved by analyzing certain Interface Control Documents (ICDs). The following ICDs were selected for analysis[1]:

- ICD 268692: Full Time Support Management Control System (FTSMCS)

[1] Since this work was done, some of these ICDs have changed and others are no longer required.

- ICD 277871: Pay Entitlement Processing Application (PEPA)
- ICD 289993: AF Recruiting Information Support System (AFRISS)
- ICD 297955: Automated Line of Duty Determination System (ALOD)
- ICD 300128: Web Intensive New Gain System (WINGS)
- ICD 314877: Centralized Disbursing System (CDS)

The ICDs contained both common and unique aspects of each airman. Hence, they are excellent candidates for transformation into an aggregate information model.

The resulting XSD contained 1400 lines of code and was considered too large for a prototype effort. However, developing this schema, provided insights into the characteristics of a comprehensive Person Business Object:

- The Person Object definition itself will be a large schema definition.
- The Person Object definition would contain major categories of data (Personally Identifying Information, Service Data, Education Data, etc.). Interface Partners generally ask for these particular kinds of data, and if the Person Object Definition could consider these categories in its structure, isolating data for a particular Interface Partner would be as easy as selecting which first or second level elements to send.
- Information exchanges require different formats for the same information (i.e. date format as YYYY-MM-DD or MM-DD-YYYY), so agreeing on a format for the Person Object definition should include all of the information required to transform the data to other desired formats.

Two particular ICD’s (FTSMCS and ALOD) were selected for the prototype effort because they were representative and used different communications protocols. The resulting XSD contained 140 lines of code.

Modeling in XML

The FTSMCS and ALOD ICD's underwent detailed analysis, with a two-fold objective: to create a representation in code of the ICD packages; and to visually examine and determine paths for transformation. The XML Schema representations were created and combined, using the <oXygenXML/> editor. An aggregate object (the "Person Object") is shown below:

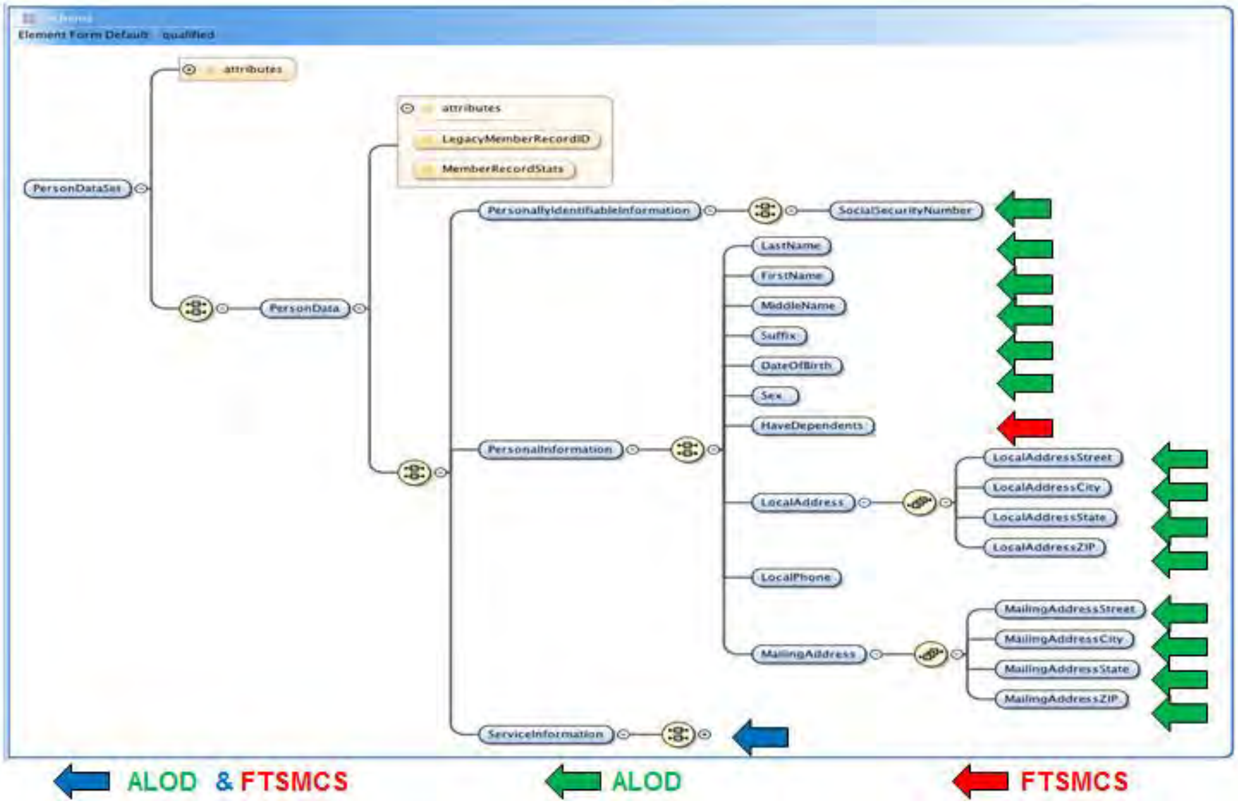

Figure 1. Snippet of Aggregate Object (oXygenXML Graphical View).

Figure 1 depicts the top level structure of the prototype Person Object as an XSD.

- The *PersonDataSet* element allows for multiple *PersonData* elements. The *PersonDataSet* element could include attributes including count of elements in the set, and a possible checksum for ensuring validity.
- The *PersonData* element will contain all of the data related to a person. Each element below the *PersonData* element will contain the major category of data.
- The Personally Identifiable Information (PII) element contains sensitive data that identifies the person.
- The *PersonalInformation* element contains information about the physical characteristics of the person and where they reside.
- The *ServiceInformation* element contains information related to their military service.
- In this example, *PersonData* is comprised of data elements from ALOD and FTSMCS. The Green Arrows identify what data elements come from the ALOD ICD while the Red Arrows identify what data elements came from the FTSMCS ICD.
- The Blue Arrows identify where both the ALOD and FTSMCS ICDs contained the same data elements. (While collapsed in this diagram, many data elements were in this category.)

This *Person Object* was utilized for the XML information exchanges for the interface strategy prototype.

Referenced AF-IPPS Architecture

The following diagram depicts the Interface Architecture Strategy that was displayed at the April 2011 AF-IPPS Industry Day at Hanscom Air Force Base.

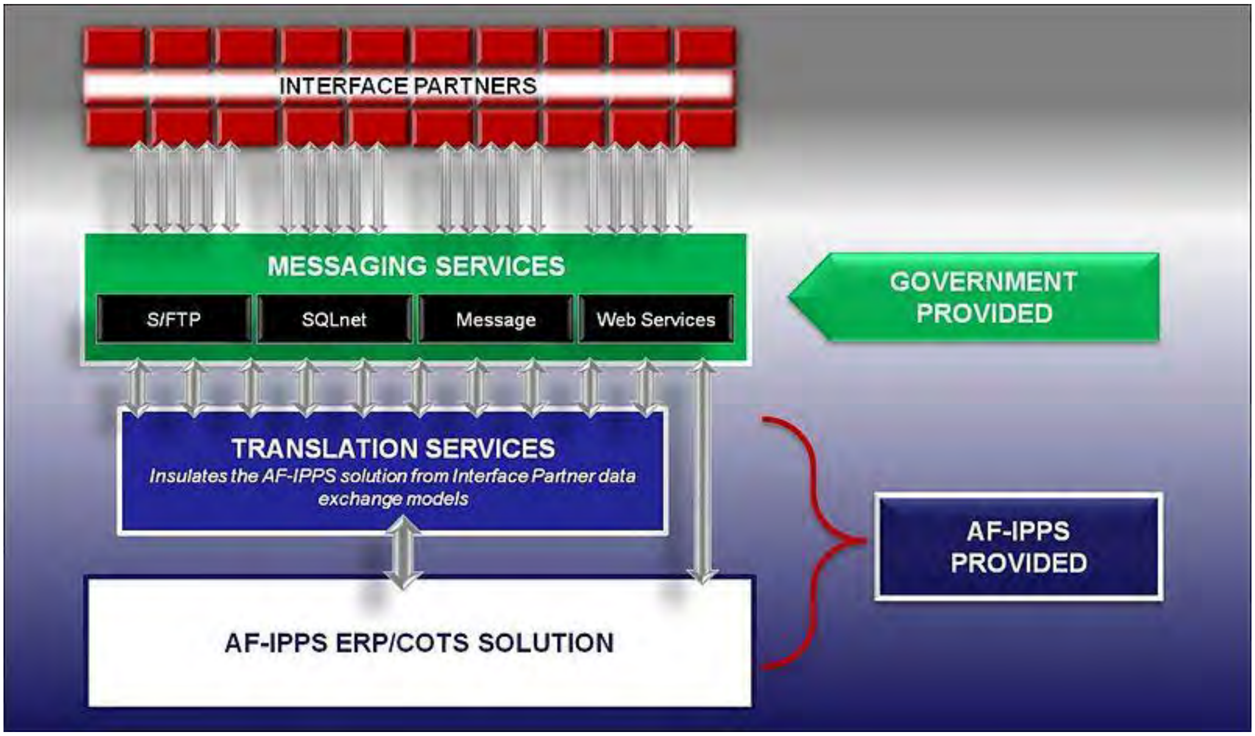

Figure 2. AF-IPPS 04/2011 Industry Day - Interface Architecture

The Interface Architecture in Figure 2 contains four major elements. The Interface Partners and AF-IPPS are the systems that wish to exchange information. The Green box represents the types of messaging used by the Interface Partners. Over time, technologies have changed, and Interface Partners exchange information in different ways. The translation services layer (Blue box) diagrammed above will perform the format and protocol transformation between AF-IPPS and the Legacy Systems. Ultimately, when an agreed upon format and protocol for the data is established, then a translation layer would no longer be required (as in the right- most arrow between AF-IPPS and the messaging services).

Software Baseline

The prototype software architecture makes extensive use of best-of-breed open source software. The product baseline is summarized in the following table:

Product	Function	Description
Java	Language	Compiler & Runtime
JBoss Application Server	Applications, Messaging, Translation	Java EE Application Server
Ruby on Rails	Applications	Ruby Web Framework
MySQL	Data Store	Database
WS02	Secure FTP, Routing	Enterprise Service Bus

Layer7 SecureSpan Gateway	Security	SOA Firewall
<oXygen/>	XML Design, Testing	XML Editor

Table 1. Software Products Used in Prototype Development

Software Architecture

The AF-IPPS Reference Architecture depicted in Figure 2 was used as the basis for the software architecture for the interface prototype, as shown below:

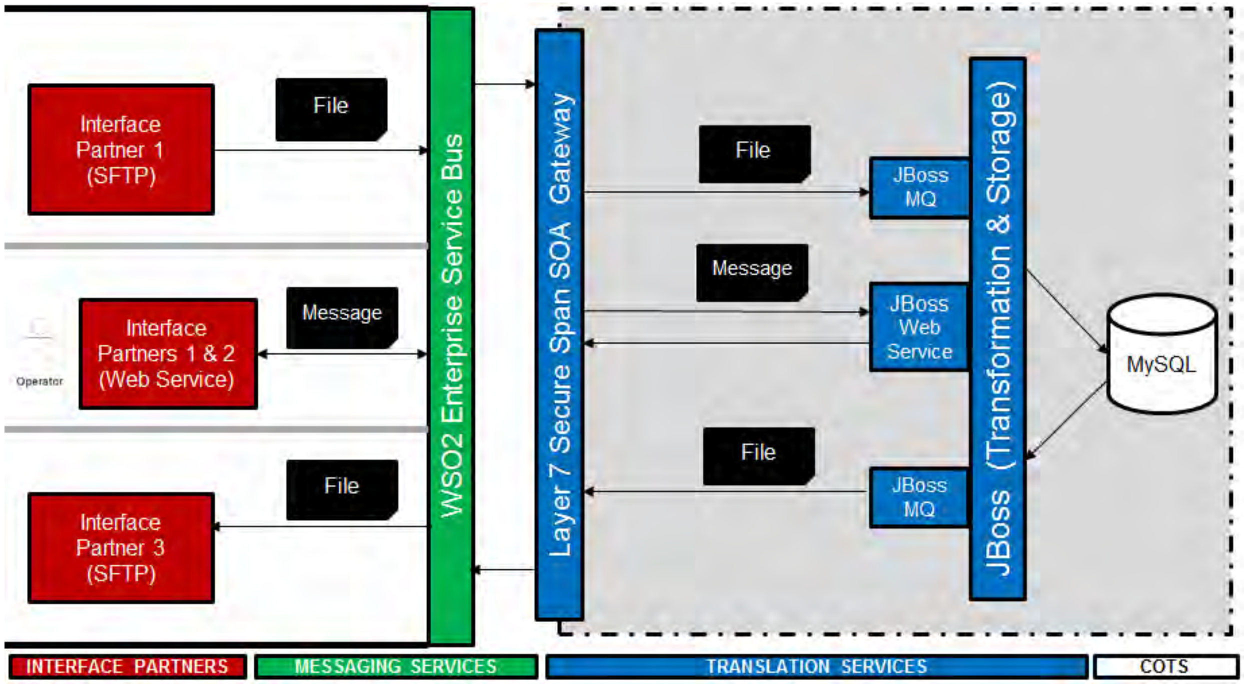

Figure 3. Interface Prototype - Software Architecture.

Core services are written as Java EE applications, running on the JBoss Application Server, which features the JBoss Web Services and Messaging/Queuing components as part of the constituent micro-kernel architecture. This provides the necessary publish/subscribe capability in addition to web-service features.

All transformation occurs in the JBoss application server at the business logic tier. Original message content and transformed data are stored in the MySQL database – which is representative of the enterprise resource planning (ERP) persistent data store.

Secured file transfer communications are processed via the WS02 Enterprise Service Bus. Client web applications were written using the *Ruby on Rails* web framework, which is specifically geared for rapid prototyping. The prototype also uses the commercial *Layer7 Secure Span Gateway* to implement the strong security layer.

Security Architecture

The Secure Span Gateway enables a strong security posture; with digital certificate (PKI) authentication, policy-based authorization and response customization.

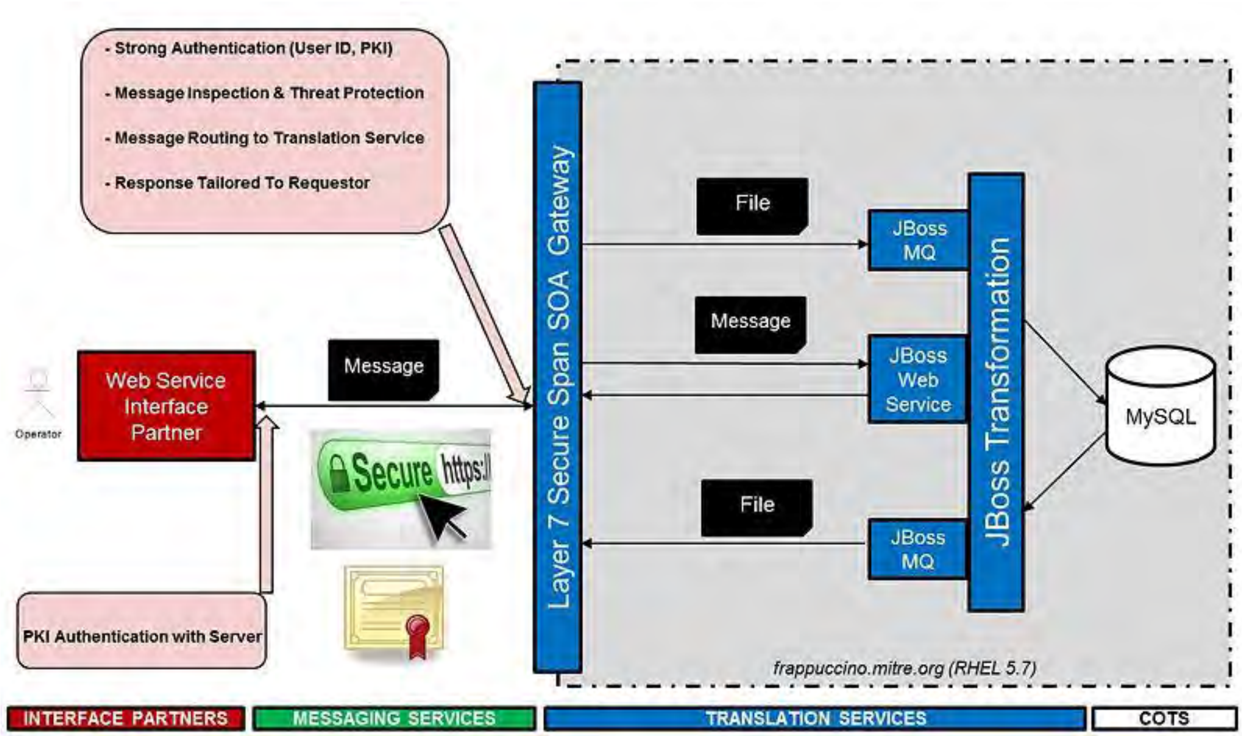

Figure 4. Interface Prototype - Security Architecture.

Transport Layer Security (TLS) ensures the communications between the client and server are encrypted. The attributes in the certificate ensure that only users who are in an allowed group can access exposed services which are abstracted from the actual system service endpoints.

Once authenticated, a request goes through authorization checks to ensure the requestor is permitted access to the requested service. Subsequently, threat protection rules execute to check for malicious code injection, document structure threats and SQL attacks.

Once the authorization checks are complete, the request is routed to the appropriate system endpoint for fulfillment. Finally, response content is customized on a per-requestor basis.

The Secure Span Gateway Policy Manager Console is shown below, with the policy definitions identified appropriately:

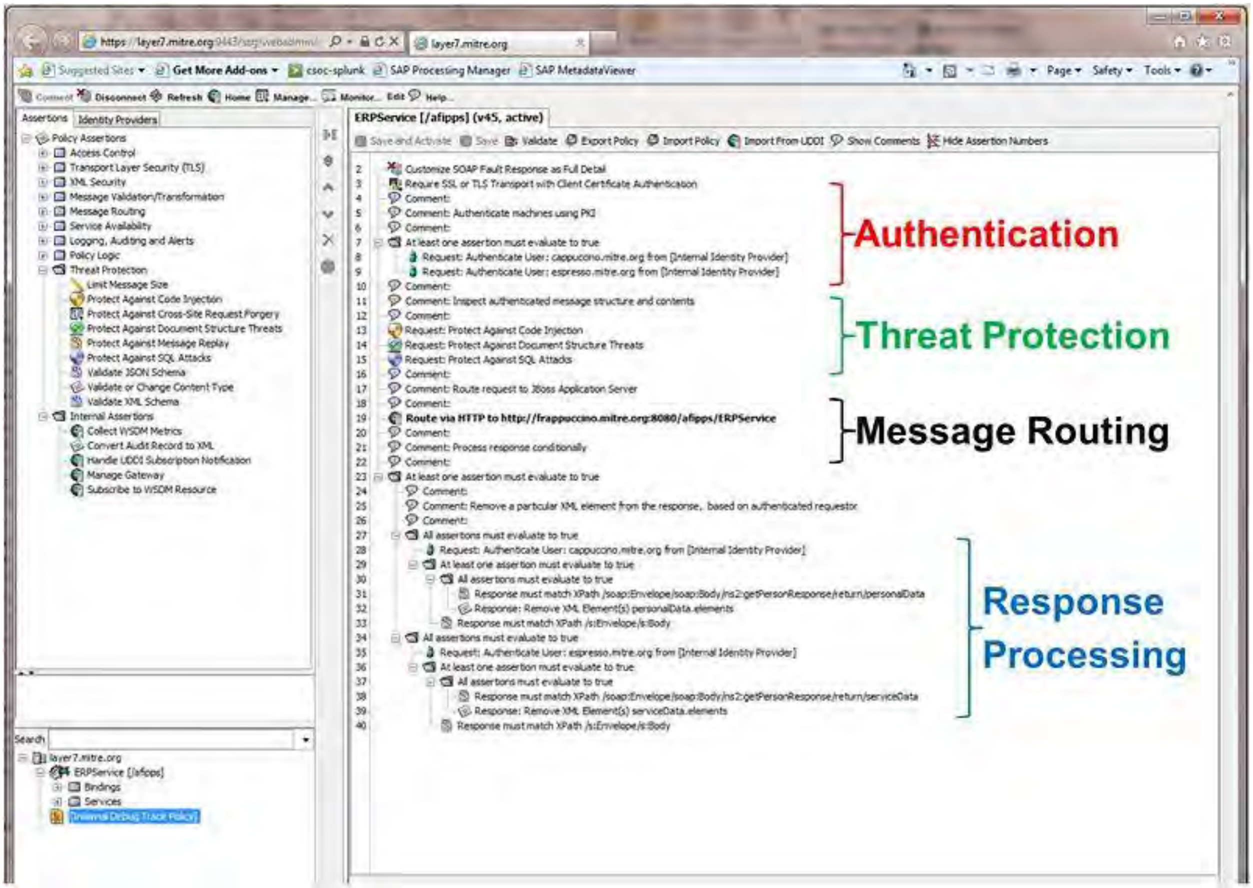

Figure 5. SecureSpan Policy Manager Administration.

Execution Flows

The prototype features two types of execution threads:

1. Asynchronous Processing Thread via Secure FTP:
    a) Inbound interface partners make plain text structured files available, containing person data that conforms to particular ICD specifications.
    b) The prototype obtains these files via SFTP and processes them, transforming the input person data into new representations.
    c) Transformed person data is then delivered asynchronously to an outbound interface partner; in plaintext structured format, via SFTP.

2. Synchronous Processing Thread via Secure Web Services:
    a) A web service client makes a secure request for transformed data.
    b) The request is fulfilled, with the response customized on a per-requestor basis.
    c) Communications are implemented both via SOAP Web Services (XML messaging); and RESTful Web Services (JSON messaging). The REST endpoints are implemented via the Layer 7 Gateway (with no change to SOAP services implemented at the JBoss level). They serve to illustrate the lightweight nature of JSON, as an alternative to XML for data exchange.

We demonstrated the prototype to the AF IPPS PMO using simulated data for ALOD and FTSMCS. The prototype demonstration showed the ability to obtain transformed data through different protocols, as described in the above SFTP and web services execution threads.

Benefits

The advantage of this approach over many point to point interfaces is that the extract and push of person data happens only one time for all interface partners. Furthermore, any changes to the translation layer are compartmentalized into one application outside of the ERP platform, so there is no impact to the ERP system for changes required by external interface partners. This results in reduced maintenance costs while also improving the performance of the transactional ERP platform. It also makes it easier to instantiate any new interfaces as they become necessary.

Findings

1. Open-source software platforms like JBoss are viable alternatives for implementing AF-IPPS.
2. Web Services & Messaging standards are sufficiently mature to support integration with ERPs.
3. The architecture supports separation of concerns as a key principle for segmenting and compartmentalizing execution flow. This concept promotes the idea of having multiple vendor products that interoperate using open standards to fulfill requirements.
4. The concept of a Person Business Object, while viable, will require analysis to efficiently select data from the large number of sub-categories of personnel data.

Conclusion

The results have demonstrated that the concept of the AF-IPPS interface translation layer is viable and can be implemented with standards-based Open Source and COTS products.
In addition, the prototype demonstrates how the translation layer can be used to insulate the AF-IPPS solution from changes in legacy systems and the issues with point-to-point interfaces can be minimized. The delivery model provides information content format and delivery protocol translation. It also aligns with the AF web-services requirement.

Although the translation layer was not tested for performance, it is built with software components that are widely used in the commercial industry and could thus be enhanced to meet specific performance requirements. For example, JBoss Application Server complies with the Java EE Specification which provides several mechanisms for enhancing performance (clustering of services, JVM tuning, etc.). Also, performance issues related to a very large person object can be handled by publishing changes only on a periodic basis or by using a pipeline processing application such as *Ab Initio* to publish the information.

References

1. Improved Management Oversight of Business System Modernization Efforts Needed, GAO-11-53, Oct 7, 2010: (http://www.gao.gov/products/GAO-11-53)

2. Air Force Policy Directive 33-1, 9 August 2012
(http://www.e-publishing.af.mil/shared/media/epubs/AFPD33-1.pdf)
3. ESC I63-1200 3 July 2008
(http://www.e-publishing.af.mil/shared/media/epubs/ESCI63-1200.pdf)
4. DoD Net Centric Services Strategy
(http://dodcio.defense.gov/Portals/0/documents/DoD_NetCentricServicesStrategy.pdf)
5. DoD Net Centric Data Strategy
(http://dodcio.defense.gov/Portals/0/documents/Net-Centric-Data-Strategy-2003-05-092.pdf)